Measuring transaction success in spoken dialogue information systems

Hans Dybkjær
dybkjaer@pdc.dk
Prolog Development Center A/S
DK-2605 Brøndby, Denmark

Laila Dybkjær
laila@nis.sdu.dk
NISLab, Southern Danish University
DK-5230 Odense M, Denmark

ABSTRACT

Usability is an important competitive parameter when a system is put to the market place. In spoken language dialogue systems transaction success is one of the frequently used metrics in deciding whether a system is acceptable according to the contract. Although there is not always any clear correspondence between user satisfaction and transaction success rate, transaction success is probably still an important contributor to the overall system usability.

This paper which reports on preliminary still ongoing work, discusses a coding scheme for annotating transactions and presents results from using it in the annotation of nearly 450 dialogues collected before and after a Danish spoken language dialogue system was put in operation.

General Terms

Design, Experimentation, Human Factors, Languages.

Keywords

Spoken dialogue structure, transactions, annotation.

1 INTRODUCTION

One of the key concerns when bringing a spoken language dialogue system (SLDS) to the market place is to ensure a high transaction success. A high transaction success means that users mostly get the correct information in return to their questions or choices. Although there is no clear one-to-one relation between transaction success and user satisfaction, transaction success is no doubt an important parameter in ensuring user satisfaction. Often a minimum transaction success is stated as a requirement in the contract for an SLDS.

This paper presents work on annotation and analysis of transaction success carried out during development of the so far most advanced Danish SLDS. The system which is called FAQ (frequently asked questions) is an over-the-phone SLDS developed for FerieKonto by Prolog Development Center (PDC) and NISLab.

In the following we briefly describe the FAQ project, including FerieKonto and their customers, i.e. the users of the system, and the system development (Section 2). Section 3 discusses dialogues and transactions. Section 4 presents data collection, annotation and results. Section 5 concludes the paper and discusses future work.

At the end of the paper there are two sample dialogues, the first is a real user of the system installed at FerieKonto talking with an first, limited version, the second a test call with the full system.

Nordtalk Symposium on Relations Between Utterances, December 5-7 2002, Copenhagen, Denmark.

2 THE PROJECT

The public institution ATP (The Danish Labour Market Supplementary Pension Scheme) hosts a number of pension-related funds, one of which is FerieKonto that administers holiday allowance for many Danish employees. ATP constantly aims at improving their service to customers while at the same time keeping the costs at a minimum or even reducing them. So far FerieKonto has via a voice-response system offered very general information on what to do if you have a holiday allowance form and on when you will get your allowance. This system is very simply with very limited domain coverage. FerieKonto also has a well-designed internet site answering typical questions, but many of their customers either don't have access to the internet or prefer to use the phone for such questions. Therefore ATP wanted to get an automatic phone system for answering typical questions. The FAQ system is developed in response to this need. More background information on the project and its partners can be found in [Dybkjær and Dybkjær 2002].

2.1 FERIEKONTO AND THEIR USERS

According to the holiday allowance law which covers all employees in Denmark, an employer must pay 12,5% of the monthly salary of an employee to an approved institution administering holiday allowance. FerieKonto administers holiday allowance for a large amount of employees. In 2001 the employers paid about 700 million Euros to FerieKonto.

Employees continuing in a non-temporary position will just get their ordinary salary during their holiday. The administration of their holiday allowance is something they don't need to bother about. However, if they change to another position, leave the labour market (pension, unemployment, leave, …) or have a temporary position, they will get a holiday allowance certificate and will get allowance during their otherwise unpaid holiday by filling in this form and submitting it to, in most cases, FerieKonto.

Thus users of FerieKonto are adults of any age up to pension age, and of both sexes. A typical user of FerieKonto will call 0-3 times in a lifetime. Therefore in reality all callers are first time users. The calls may concern person related issues, such as "Have you transferred the holiday allowance I asked for", "How much money/many days do I have in my account?", or the questions may be of a general nature that may be categorised as typical questions, such as "Is Saturday a holiday", "Can I get a new holiday allowance certificate", "I didn't have time for holiday, can I get my money anyway".

2.2 THE FAQ SYSTEM

The FAQ system can answer typical non-personal questions, i.e. it can provide guidance on how and when a person can get his holiday allowance depending on this persons situation (being an ordinary employee, having moved abroad, being on leave, etc.). In addition it can inform on related topics such as the address of FerieKonto and when the holiday year starts.

A two-step approach has been used during development of the FAQ SLDS. We knew that the FAQ system would be a very challenging system to develop with its unstructured task domain which is difficult to handle in a reasonable way without adding many annoying dialogue constraints. At the same time we also knew that there were several potential problems ahead of us on the technological side: the Philips SpeechMania platform on which the system is developedwas new to us and not least was the telephony side something on which we had no previous experience. Therefore a very reduced version of the FAQ system was developed first. This small system was called Vejled (Guidance) and its primary goals were (i) to get the technology into place while still having a relatively simple dialogue system, and (ii) to generate initial experience with real users. The second step was to enhance Vejled into a real FAQ system.

Vejled was put into operation by the end of August 2002. Since then focus has been on the full-blown FAQ system and on collecting experience from real customer dialogues with the Vejled system. FAQ is now in its final phase and will be put into operation by medio December 2002. Section 7 shows two example dialogues collected with the Vejled system and the full FAQ system, respectively.

The FAQ system has a number of distinguishing features compared to most current commercial systems:

- The number of issues it knows about is large. Our current version runs with 80 concept slots (75 task topics and 5 meta topics).
  For comparison, a typical travel system has origin, destination, date, starttime, endtime, and meta yes/no/help/repeat, that is about 10 concepts.
- The users do not know the domain, and they often do not know how to formulate their problem. In a typical SLDS users know the domain and their problem very well.
- There is no clear task. One has to negotiate and make a conversation in order to find the right information.
  Typical systems have a clearly defined goal where certain parameters must be filled (and both the system and the users know that).

In order to solve this we have carefully designed a dialogue where users are offered a choice between user-initiative and system guidance. Users may take the control themselves and try with their own words to jump directly to relevant information areas, or they may let the system guide them through the main areas. Also, when information has been given, the system will offer related information.

Other system features include graduating feedback according to recognition scores, letting output depend on what the user has heard before, and making variations of the most frequent politeness and query phrases.

3 DIALOGUES AND TRANSACTIONS

Transaction success measures the number or percentage of completed transactions. It is an objective measure but there is no standard definition of what is a transaction. What constitutes a transaction depends on the type of task dialogue, including:

- Single task dialogues (postal information, traffic information, flight booking, …).
  The entire dialogue will often only contain one transaction, e.g. that of booking a ticket. If there are more transactions they will concern the same task, e.g. a user may query about two different train connections. Most commercial dialogue systems belong to this category. There is a small, well-defined number of fields that need to be filled in, and both the caller and system know which they are. Users normally have a clear goal in mind when they call, such as getting information on train connections between two cities in the morning, or booking two flight ticket to a particular destination. One may talk about sub-transactions, such as determining a date or a destination, but they are not independent and only serve to contribute to the overall transaction.
- Composite task dialogues (faq, portals, …). There may be many, different transactions in each dialogue. The number of concepts is large, only few of them need to be filled in before task

completion can be attempted, and users need not have a clear picture of the task domain and of what exactly they want to achieve before calling. The present paper concentrates on composite information task dialogues.

- Non-task dialogues (e.g., leisure where conversation itself is a target). The notion of ‘transaction’ is at best unclear, and maybe not applicable.

The extent to which we can analyse transactions depends on the information available. In our case we have the log information produced by the platform SpeechMania (™ Philips Speech Processing, now ScanSoft). This includes textual representation of system output, recognised input, user sound input, and any additional application printouts, such as time stamps or recognition scores. It does not include system sound output or precise information on barge-in, i.e. in case of barge-in we do not know precisely how much of the system output has been played to the user. The system runs with barge-in, so in the very moment the user says something the system will interrupt its output and interpret the user input according to the next user input point in its dialogue model.

3.1 DIALOGUE TERMS AND PATTERNS

We will model dialogues according to a turntaking model based on utterances where each utterance has a number of attributes, including generic domain, topics, acts and interlocutor (‘who’).

A *turn* is defined as a maximal sequence of non-interrupted utterances by the same interlocutor.

At the transcription level a dialogue is a sequence U* of utterances subdivided into turns. An *utterance* is an arbitrary sequence of words and has the following attributes:

- *interlocutor*: In our case it is either *user* or *system*. We use ‘user’ and ‘caller’ as synonyms. In some calls one may hear a third person communicating with the user, typically on the dialogue and what to do. So far we have disregarded this phenomenon.
- *generic domain D*: Any utterance may concern a specific task, concern the dialogue itself (meta), or concern other stuff, cf. Figure 1.
- *topic T*: A topic basically is the same as a *concept* in the sense of the SpeechMania dialogue programming language HDDL. However, whereas the HDDL concept is an implementation device, a topic is an analysis device. An utterance can have more than one topic.
- *acts A*: Every utterance has one or more dialogue acts, cf. Figure 2. There are many other possible sets of dialogue acts, with other boundaries and granularities, but these are the ones used in this document.

The notation DA(T*) will denote an act A in (optional, default is task) domain D concerning the topics listed in the sequence T*. The sequence may be empty, e.g. “Goodbye” becomes o().

With these definitions in place we can formalise patterns of dialogue. We will let <s attributes> and <u attributes> denote utterances for the system and the user respectively. Some pattern examples are given in Figure 3, where we use common notation (e’|e”) for alternatives, and e* for zero or more occurrences of e. One can use these patterns when designing reusable dialogue consistent behaviour. In this document we will use them to support the discussion of transactions in the next section.

Figure 1: Generic domains.

Shorthand	**Generic domain D**	**Example**
t	Task	Email fk@atp.dk
m	Meta	Sorry,
o	Other	Goodbye

Figure 2: Dialogue acts.

Short	**Dialogue act A**	**Example**
q	Offer/question	*Should I repeat the address?*
i	Information	*Email fk@atp.dk*
f	Feedback	*If you are an employee…*
a	Accept	*Yes*
r	Reject	*No thanks*
s	Selection	*Employee*
o	Other	*Where is my hat*

Figure 3: Some common dialogue patterns.

Description	**Pattern**	**Example**
System initiative	(<s (i|f|o)*> <s q> <u (f|a|r|s|o)*>)*	*Hello. Say 'zip' or 'postage'.* *Zip.*
User initiative	<s q(T*)> <u s(T, T not-in T*)>	*Say if you are A, B or C* *What about D?*
Implicit feedback	<u s(T)> <s f(T)i(T')>	*I am an employee* *If you are an employee, then …*
Explicit feedback I	<u s(T)> <s f(T)q(T)>	*Employee* *Did you say employee?*
Explicit feedback II	<u s(T)> <s f(T)> <s mq()>	*Employee.* *Employee.* *Is that right?*

3.2 TRANSACTIONS AND THEIR ANNOTATION

Based on Section 3.1 we will now discuss transactions and how to calculate the transaction success. Since FAQ dialogues are composite task dialogues one dialogue may contain several transactions as explained above. A transaction usually spans at least one user utterance and one system utterance and often more.

An utterance may concern any of the three generic domains listed in Figure 1. An utterance is categorised as belonging to the 'other' domain if it is neither 'task' domain related nor 'meta' domain related, e.g. "Hand me the butter, please" in a conversation on holiday money. Also, data shows that many people try to close human-computer conversation by explicitly saying 'goodbye', not realising that when talking to a machine hanging up is not impolite. 'goodbye' also belongs to the domain we have termed 'other'.

An utterance belongs to the meta domain if it concerns the dialogue itself rather than a particular topic. A typical example is an utterance in which the user tries to repair a system misunderstanding. The fewer transactions in the meta domain the more smooth the dialogue.

An utterance concerning a particular topic belongs to the task domain. Normally most transactions in a dialogue concern the task domain and are closely bound to topics. In the Vejled system there are only 14 possible topics, e.g. address, phone, stopped working due to health or age, and live abroad. Getting the information related to one topic corresponds to one transaction.

However, in the FAQ system there are 75 topics, and many of these may be combined to point out sub-specialised information.

In principle one could define a transaction simply in terms of its start and end where the end may be either a success, a failure or wrong (see further explanation below). However, this does not provide much information about exchange patterns and potential problem patterns. Thus we want to break down the transactions into smaller components. To this end we can use the dialogue acts listed in Figure 2. Each of these dialogue acts may form part of a transaction and we then arrive at the following break-down of transactions.

A transaction is always something the user decides to start. Basically a transaction may be initiated by

- the user actively *starting* a new topic via a question, or
- the system *offering* one or more topics and the user *accepting* it or, if more than one, *selecting* one.

A transaction continues if

- the system asks *clarifying* questions and the user answers these,
- the system *checks* the correctness of what it understood by asking the user, and the user confirms, or
- the system informs on a *wrong* topic but the user then produces a *repair* utterance.

Checking the correctness of what has been understood corresponds to providing explicit feedback.

If the system does not understand a start or an accept correctly, the user may initiate a repair dialogue. The repair dialogue is considered part of the transaction to which the repair relates. Similarly the system may initiate repair if it does not hear or understand the user's input.

A transaction is closed either by

- the system *successfully* providing the correct requested information,
- the system *failing* to provide the requested information, or
- the user *rejecting* one or more topics offered by the system which means that the transaction is never really started.

If the system provides information rejected by the user we shall call this transaction end wrong. If the system does not provide the information accepted, selected or asked for but if the user tries to repair the dialogue we shall call this intermediary transaction end wrong. If the user with or without repair fails to get the information he wants, the last system utterance belonging to the transaction is called a failure.

If the system actually delivers the information accepted or asked for with or without repair exchanges, the last system utterance belonging to the transaction is called a success.

Some cases will not be counted as transactions:

- the user provides *other* input, i.e. input which is not a meaningful contribution to the dialogue,

- a dialogue does not contain any (meaningful) user input at all and should therefore be *discarded*.

Sometimes users play with the system and provide non-sense input such as letting their canary bird sing or calling out the name of the woman whose voice is being used for output. Such input we will categorise as an “other” act. The subsequent system reply is not counted as (part of) a transaction since the actual answer to such non-sense input is not interesting as long as the system does not break down. Note that it is a matter of interpretation to decide if something is meaningful in the context. Some out of task domain questions may still be meaningful because they are related to the domain covered by the system. In such cases input should not be categorized as an “other” act.

Some of the annotated calls are only test calls in some sense where a developer or a user has called the system, heard it say something, and then just hung up again without entering into a dialogue. Such dialogues have been excluded (discarded) from the material on which we measure transaction successes and failures.

Based on the above break-down of the constituents of a transaction and in order to mark up transactions we created the tag set shown in Figure 4. Two tags had to be added to the original tag set used on Vejled dialogues when annotating transactions in FAQ dialogues. Note that the coding scheme using this tag set is a cross-level coding scheme using tags both from the level of dialogue acts and from the level of pure transactions, cf. [Dybkjær et al. 1998].

Figure 4: Tags used for mark up of transactions.
The two tags ‘check’ and ‘select’ were added when tagging the FAQ dialogues.

Tag	Explanation	Type	Application
accept	User accepts system offer.	Act <a>	faq + vejled
repair	User/system rejects or corrects action done by system/user.	Act <mf><q>	faq + vejled
other	User makes unclear or null action.	Act <o>	faq + vejled
offer	System offers information to user.	Act <q>	faq + vejled
reject	User rejects system offer.	Act <r>	faq + vejled
select	User selects from offer list	Act <s>	faq
check	System makes explicit confirmation	Act <tf> <q>	faq
discard	Call should be discarded with respect to this scheme.	No type	faq + vejled
fail	Previous transaction ends with failure.	Transaction	faq + vejled
start	User initiates new task = request for information.	Transaction	faq + vejled
success	Previous transaction ends with success.	Transaction	faq + vejled
wrong	System responds with wrong topic.	Transaction	faq + vejled

Building on the above transaction elements and the dialogue patterns of Figure 3, we can establish patterns of transaction success. Figure 5 provides examples of such patterns. The “success” column indicates if the transaction is complete and if it is a success. Such patterns serve to give a precise definition of when a transaction can be termed a success.

It is important for an annotator to know when a transaction is considered a success and when it is a failure to ensure high coding quality. Once a batch of dialogues has been annotated with respect to transactions, the success rate can be calculated as: #success / (#success + #fail).

In the following section we describe data collection with the Vejled system and the annotation with respect to transactions of some of the collected dialogues. We also present preliminary results, including the transaction success rate.

Figure 5: Transaction success patterns (not complete).

Description	**Pattern**	**Example**	**Success**
Smooth transaction (but no feedback)	<s q(T*)> <u s(T, T in T*)> <s i(T)>	*Sig om du er lønmodtager, …* *Lønmodtager* *Så skal du …*	yes
User initiated transaction (but no feedback)	<s tq(T*)> <u ts(T, T not-in T*)> <s ti(T)>	*Sig om du er …* *Hvad er jeres adresse* *Feriekonto, Kongens Vænge 8, …*	yes
System clarification in transaction	<s q(T*)> <u s(T, T in T*)> <s f(T) tq(T’*)> <u a()> <s f(T,T’)> <s i(T)>	*sig om du er …* *frameldt* *Bor i udlandet. Arbejder du i Danmark* *ja* *Hvis du bor i udlandet, men arbejder i Danmark* *så gælder …*	yes
System miscommunication, recovered after user initiated repair	<s q(T*)> <u s(T, T in T*)> <s f(T’) i(T’)> <s mq()> <u r() s(T)> <s f(T) i(T)> <s mq(again)>	*sig om du er …* *forladt. Hvis du er lønmodtager, så …* *Vil du have det gentaget* *nej, jeg har forladt arbejdsmarkedet* *Hvis du har forlad …, så …* *Vil du have det gentaget*	yes Counts as one large transaction
System initiated meta-communication I	<s q(T*)> <u x()> <s mf()> <s q(T*)> <u s(T, T in T*) <s f(T) i(T)>	*Sig om du er …* *Jeg er* *Undskyld, jeg forstod ikke hvad du sagde* *Sig om du er …* *Jeg er …* *Hvis du er … så skal du …*	yes
System initiated meta-communication II	<s q(T*)> <u s(T, T not-in T*)> <s mf(X)> <u s(T, T not-in T*)> <s f(T) i(T)>	*Sig om du er …* *Jeg er studerende* *Sagde du minister?* *Studerende* *Hvis du er studerende, så …*	yes
Question for existing subtask	<s q(T)> <u a()> <s i(TT’)> <s mq(again)> <u s(T’)> <s i(T’)>	*Vil du høre adressen* *Ja* *Feriekontos adresse er … email …* *Vil du have den gentaget* *Hvad var jeres email* *email …*	yes. Only one transaction
Question for new subtask	<s q(T)> <u a()> <s i(TT’)> <s mq(again)> <u s(T”)> <s i(T”)>	*Vil du høre adressen* *Ja* *Feriekontos adresse er … email …* *Vil du have den gentaget* *Hvad er jeres telefonnummer* *Telefon …*	yes. And then a second, different transaction: yes.

4 DATA COLLECTION AND ANNOTATION OF TRANSACTIONS

4.1 DATA COLLECTION

We have so far collected a few thousand Vejled calls and about 500 FAQ test calls. In this paper we concentrate on three Vejled test batches and one Vejled operation batch. Each set of dialogues was transcribed right after the end of the period during which it was collected. Transcription was done using the SpeechMania transcription station.

From 21 March to 8 May 2002 we collected 225 test calls to the Vejled system. For practical reasons the calls have been split into three sets.

Set 1 contains 50 dialogues recorded between 21 March and 4 April. Callers were people from or closely related to the development group, and the dialogues tend to be somewhat experimental.

Set 2 contains 104 dialogues recorded between 5 April and 23 April, and set 3 consists of 71 dialogues recorded between 24 April and 8 May. Most callers in these two sets were unknown to the development group. Those people had not tried the system before and the knowledge they had about the system would either come from a web page briefly describing the system and its functionality and containing a questionnaire, or from a sheet of paper with the same information. Colleagues, friends and family members to the person in charge of carrying out evaluation of the system were equipped with a reference to the mentioned web site and usually also with a number of paper sheets with the corresponding information. They were asked to invite whoever they wanted and as many as possible to try the system. It should be noted that in all cases the calls came from people who were invited to test the system. They did not have a real, personal problem to ask about.

By the end of August 2002 Vejled was put into operation. The automatic collection of dialogues still continues. Approximately every week the batch of dialogues collected during this period is sent for transcription. We have selected a batch of transcribed dialogues from the last week of September containing 217 calls. These calls were made by real users calling FerieKonto because they had a question to which they wanted an answer.

All 225 plus 217 calls were annotated by one and the same person who is familiar with corpus annotation using the tool described in [Dybkjær and Dybkjær 2002b]. The tag set explained above was used for marking up the dialogues.

4.2 TRANSACTION RESULTS

Figure 6 shows the occurrences of tags in each of the four sets of dialogues mentioned above, including the number of transaction successes and failures. The figure also shows the total number of calls per set, number of calls with at least one transaction failure and the transaction success rate in percent. Finally, the number of so-called smooth calls per set is shown. A smooth call is here defined as a non-discarded dialogue with no transaction failures.

Figure 6: Tags and their occurrences in the four sets of dialogues.
Transaction success percent = #success/(#fail+#success)*100.
Smooth call percent = (#calls-#discard-#calls with at least one fail)/ (#calls-#discard)*100.

Vejled dialogues	**Test sessions**				**Operation**
Tag	**Set 1** **2002-04-04**	**Set 2** **2002-04-23**	**Set 3** **2002-05-08**	**Total**	**Set A** **2002-10-01**
accept	30	38	10	78	37
discard	13	41	34	88	118
fail	18	8	10	36	19
offer	99	125	84	303	231
other	16	31	14	61	43
reject	37	75	50	162	77
repair	22	11	12	45	44
start	142	139	72	353	120
success	153	168	81	402	133
wrong	26	6	9	41	18
Calls with at least one fail	**11**	**8**	**9**	**28**	**19**
Total number of calls	**50**	**104**	**71**	**225**	**217**
Transaction success percent	**89.5**	**95.5**	**89.0**	**91.8**	**87.5**
Smooth call percent	70.3	87.3	75.7	79.6	80.8

The transaction success rate was on average higher in the test sets than in the production set. If we look at the reasons for the failures, there is a significant difference between the test sets and the production set. In the test sets most failures were caused by a few problems in the dialogue model which still needed correction and by a sub-optimal language model. The language model and the dialogue model have been improved in the period between the test session sets and the operation set. However, users make a difference. In the test sessions users did not have a real problem they wanted help with, so they tended just to let the system lead them through the dialogue and do nothing more than that. In contrast some of the real users tried to present their actual question finding that the system by itself did not offer the information they were looking for. Thus in the production set most failures were due to users asking questions about topics not included in Vejled. The good news is that these questions all related to topics included in the FAQ system.

5 CONCLUSIONS AND FUTURE WORK

The presented coding scheme for annotating transactions has worked well for Vejled dialogues, and based on initial tests we believe that it generalises well to the in every aspect more complex FAQ dialogues. As mentioned in Section 3 we have only found a need for adding two additional tags for the markup of FAQ dialogues.

Only one coder annotated the four sets of dialogues reported on in Section 4. However, to evaluate the coding scheme and coding quality we are right now testing intercoder agreement, i.e. we compare the results produced by different coders. The common measure of coder agreement is *kappa*:

$$\kappa = \frac{P(A) - P(E)}{1 - P(E)}$$

where *P(A)* is the proportion of times that the coders agree and *P(E)* is the proportion of times that they are expected to agree by chance. However, there is no sound interpretation of which kappa values are good enough, and moreover kappa presupposes independent codes which certainly is not the case neither for dialogue acts (though e.g. Carletta et al. [1997] advocate precisely this use) nor for transaction success.

A preliminary inspection of two different coders indicates that there is confusion between “fail” and “wrong”, and uncertainty on what to do in case of multi-choices (the offer-accept/reject pair was introduced to handle binary choices). But it seems that the individual coder is consistent over time.

Regardless of coding quality being good or bad, the presented coding scheme and its annotation results also have at least the following weaknesses and restrictions:

- Annotation is time consuming. If one could semi-automatise the process it would be a big help. This would also remove some of the uncertainty there is in manual annotation. However, this is a difficult task, not least as regards the user utterances the form of which cannot be predicted.
- There is more to perceived quality than transaction success, see e.g. [Walker 2002]. A high transaction success rate does not necessarily guarantee high user satisfaction. A number of soft factors such as “Is the voice pleasant?”, “Is the system efficient?”, “Does the user feel in control?” also contribute to user satisfaction. Thus if the goal is to measure the usability of a system it is not sufficient to focus on transactions only. User satisfaction is not necessarily achieved by technically excellent systems and cannot be sufficiently measured through objective evaluation. Subjective evaluation techniques, such as questionnaires and interviews,

are needed as well. Answers may be plotted into a Likert scale. The difficulties with questionnaires and interviews concern which questions to ask and how, and how to interpret the answers received [Bernsen et al. 1998].

We will briefly discuss a metrics related to transactions and efficiency, and probably contributing to perceived quality, i.e. the smoothness metrics (see Section 4.1). For the moment we consider as future work either to extend the existing transaction coding scheme or to make a second coding scheme and also annotate the dialogues using this new scheme.

5.1 SMOOTH CALLS

The transaction coding scheme is designed to handle transaction success. Smooth calls is something which the statistics can also be used to calculate but the coding was not meant for this use which means that further elaboration is needed to get a precise overview of problems and their causes and seriousness.

If a call contains a fail and a success on the same topic then the user got what he wanted but this does not appear from the figures in Figure 6. Moreover, a fail can be more or less serious but degrees of seriousness are not included in the tag set of the coding scheme used. Markup of seriousness would have to cover the following cases:

- There must be a distinction between wrong and erroneous information. By wrong information we mean a clearly incorrect reply to the user's request for information. By erroneous information we mean a reply which seemingly is correct but which contains an error such as e.g. an incorrect telephone number or a wrong date. Erroneous information is clearly unacceptable and must be corrected in the dialogue model.
- If the user gets wrong information this may be more or less serious. If he does not get what he wanted and this is evident, e.g. fax instead of email then this is not so serious compared to the user getting information which could be mistaken to be what he wanted, e.g. fax instead of phone. Even if the system says "fax number" then the inattentive user may ignore this and just write down the number assuming this is the telephone number because he asked for the phone number and the provided number has the right format.
  Wrong information which the user may mistake to be correct information if he is just a bit inattentive is in effect close to erroneous information, but as the cause is miscommunication it is much more difficult to correct in the dialogue model, if at all possible. Sometimes improvements to the language model may help.
- Erroneous information may sometimes be of a kind which the attentive user may detect is erroneous, e.g. the statement that it is Friday tomorrow. In other cases the user has clearly no chance to reveal the error during the dialogue, e.g. a wrong digit in a phone number.
- If the system misunderstands a yes for a no and e.g. closes the dialogue, then this is of course a wrong action and should be marked as such, but it is less serious than providing erroneous information and certain types of wrong information.

The smooth call percent in Figure 6 is a very rough measure since it only looks at whether there is one or more transaction failures in a dialogue or not. Seriousness, number of repairs and whether the user succeeded in getting the information he wanted after a fail are parameters which are not taken into account. An extended or additional coding scheme would have to be used to make a more elaborate calculation of smooth calls.

6 REFERENCES

[Bernsen et al. 1998] , Niels Ole Bernsen, Hans Dybkjær, and Laila Dybkjær: *Designing Interactive Speech Systems*. Springer, 1998.

[Dybkjær et al. 1998] Laila Dybkjær, Niels Ole Bernsen, Hans Dybkjær, David McKelvie and Andreas Mengel: *The MATE Markup Framework*. MATE Deliverable D1.2, November 1998.

[Dybkjær and Dybkjær 2002a] Hans Dybkjær and Laila Dybkjær: *Experiences from a Danish spoken dialogue system*. Second Danish HCI Research Symposium 2002, November 7, 2002, Copenhagen, Denmark.

[Dybkjær and Dybkjær 2002b] Hans Dybkjær and Laila Dybkjær: *A web-based annotation tool*. Nordtalk Symposium on Relations Between Utterances, December 5-7 2002, Copenhagen, Denmark.

[Carletta et al. 1997] Jean Carletta, A. Isard, S. Isard, J. C. Kowtko, A. Newlands, G. Doherty-Sneddon, A. H. Anderson: *The reliability of a dialogue structure coding scheme*. Computational Linguistics, 23 (1), 1997, 13-31.

[Giachin 1996] Egidio Giachin: Spoken language dialogue. In Cole et al. 1996, Chapter 6.4.

[Simpson and Fraser 1993] A. Simpson and Norman M. Fraser: Black box and glass box evaluation of the SUNDIAL system. In *Proceedings of Eurospeech'93*, Berlin, 1993, 1423-1426.

icsslp, walker

[Semler 2001] Steven Semler: *An overview of content analysis*. Practical assessment, research and evaluation, 7(17). Available online: http://ericae.net/pare/getvn.asp?v=7&n=17. 2001.

icsslp, walker

[Cohen 1960] J. Cohen: *A coefficient of agreement for nominal scales*. Educ. Psych. Meas. 20, 1960, 37-46.

7 Examples

7.1 Vejled version=2002-10-01 channel=0 run=0000 call=0000

System 0.1 **- Velkommen til FerieKontos talebaserede vejledningssystem.**
- Her kan du få standardoplysninger om udbetaling af feriepenge.
- Du kan altid bede om hjælp eller gentagelse.

System 0.2 **- Sig om du**
- er lønmodtager eller arbejdsløs i Danmark
- er frameldt folkeregistret og bor i udlandet
- eller har forladt arbejdsmarkedet på grund af alder eller helbred.
- PAUSE5000

User 0.1 O ***jeg er lønmodtager***

transaction 0.1.1 --- Type:start --- SubType:1 --- 2002-10-06/ld

transaction 0.1.2 --- Type:start --- SubType:1 --- 2002-11-27/mwk

System 1.1 **- Hvis du er lønmodtager i Danmark eller arbejdsløs og har et FerieKonto-bevis**

System 1.2 **- skal du udfylde det og sende det til os.**
- Udover dig selv skal din arbejdsgiver eller din a-kasse eller socialforvaltningen skrive under.
- PAUSE200

System 1.3 **- Hvis FerieKonto-beviset er korrekt udfyldt når vi modtager det, overfører vi feriepengene til din konto.**
- Ekspeditionstiden er cirka fem arbejdsdage.
- Du kan dog tidligst få dine feriepenge en måned før din første feriedag.
- PAUSE1000

transaction 1.3.1 --- Type:success --- SubType:1 --- 2002-10-06/ld

transaction 1.3.2 --- Type:success --- SubType:1 --- 2002-11-27/mwk

System 1.4 **- Vil du høre informationen igen?**
- PAUSE5000

transaction 1.4.1 --- Type:offer --- SubType:2 --- 2002-10-06/ld

transaction 1.4.2 --- Type:offer --- SubType:2 --- 2002-11-27/mwk

User 1.1 O ***nej tak***

transaction 1.1.1 --- Type:reject --- SubType:2 --- 2002-10-06/ld

transaction 1.1.2 --- Type:reject --- SubType:2 --- 2002-11-27/mwk

System 2.1 **- Vil du høre FerieKontos adresse?**
- PAUSE5000

transaction 2.1.1 --- Type:offer --- SubType:3 --- 2002-10-06/ld

transaction 2.1.2 --- Type:offer --- SubType:3 --- 2002-11-27/mwk

User 2.1 O ***nej tak***

transaction 2.1.1 --- Type:reject --- SubType:3 --- 2002-10-06/ld

transaction 2.1.2 --- Type:reject --- SubType:3 --- 2002-11-27/mwk

System 3.1 **- Vil du tilbage til starten af den generelle vejledning?**
- PAUSE5000

transaction 3.1.1 --- Type:offer --- SubType:4 --- 2002-10-06/ld

transaction 3.1.2 --- Type:offer --- SubType:4 --- 2002-11-27/mwk

Event 1 **nothing recorded**

System 4.1 **- Undskyld.**
- Jeg hørte ikke hvad du sagde.
- Sig 'ja' for at starte forfra, 'farvel' for at forlade systemet.
- PAUSE5000

transaction 4.1.1 --- Type:repair --- SubType:4 --- 2002-10-06/ld

transaction 4.1.2 --- Type:offer --- SubType:5 --- 2002-11-27/mwk

User 4.1 O ***nej tak***

transaction 4.1.1 --- Type:reject --- SubType:4 --- 2002-10-06/ld

Dette "nej tak" burde opfattes som "farvel".

transaction 4.1.2 --- Type:reject --- SubType:5 --- 2002-11-27/mwk

System 5.1 **- Undskyld.**
- Systemet kan give information om udbetaling hvis du er lønmodtager eller arbejdsløs, er frameldt folkeregistret eller har forladt arbejdsmarkedet på grund af alder eller helbred. Du kan også bede om FerieKontos adresse.
- Sig en af disse muligheder hvis du vil fortsætte, eller sig 'farvel' for at slutte.
- PAUSE5000

transaction 5.1.1 --- Type:repair --- SubType:4 --- 2002-10-06/ld

transaction 5.1.2 --- Type:offer --- SubType:6 --- 2002-11-27/mwk

Event 2 **nothing recorded**

System 6.1 **- Farvel og tak.**

transaction 6.1.1 --- Type:success --- SubType:4 --- 2002-10-06/ld

Event 3 **hangup**

7.2 FAQ VERSION=2002-11-14 CHANNEL=3 RUN=0004 CALL=0003

System 0.1 **- PAUSE1000**
- Velkommen til FerieKontos automatiske vejledning.
- Du kan lade systemet foreslå emner eller du kan forsøge med dine egne ord.
- Du vælger emner ved at tale til systemet.
- Vælg mellem emnerne:
- PAUSE200
- 'lønmodtager'.
- PAUSE200
- 'På orlov'.
- PAUSE200
- 'bor i udlandet'.
- PAUSE200
- Eller 'forladt arbejdsmarkedet' på grund af 'alder' eller 'helbred'.

- PAUSE500
- Sig et emne, sig 'gentag' eller bed om 'flere muligheder'?

transaction 0.1.1 --- Type:offer --- SubType:1 --- 2002-12-01/¨hd

User 0.1 O ***jeg er studerende***

transaction 0.1.1 --- Type:select --- SubType:1 --- 2002-12-01/¨hd

System 1.1 **- Spurgte du om**
- Hovedmenu

transaction 1.1.1 --- Type:check --- SubType:1 --- 2002-12-01/¨hd

User 1.1 O ***studerende***

transaction 1.1.1 --- Type:repair --- SubType:1 --- 2002-12-01/¨hd

System 2.1 **- Hvis du er studerende og modtager SU, kan du godt få dine feriepenge udbetalt selv om du ikke holder fri fra studierne.**
- Men hvis du har arbejde ved siden af studierne, skal du holde fri fra dit arbejde for at få feriepengene udbetalt.

System 2.2 **- Hvis du har arbejde er det din arbejdsgiver, der skal underskrive dit FerieKonto-bevis, ellers er det socialforvaltningen i din kommune.**

transaction 2.2.1 --- Type:success --- SubType:1 --- 2002-12-01/¨hd

System 2.3 **- PAUSE500**

System 2.4 **- Er der andet du gerne vil spørge om?**

transaction 2.4.1 --- Type:offer --- SubType:2 --- 2002-12-01/¨hd

User 2.1 O ***skal jeg skrive lørdage på***

transaction 2.1.1 --- Type:start --- SubType:3 --- 2002-12-01/¨hd

System 3.1 **- Fra ferieåret 2002, der starter 1. maj, er ferieugen ændret til 5 feriedage, og lørdage skal således ikke længere tælles med i ferien.**

transaction 3.1.1 --- Type:success --- SubType:3 --- 2002-12-01/¨hd

System 3.2 **- Vil du vide noget om 'ferieåret'?**

transaction 3.2.1 --- Type:offer --- SubType:4 --- 2002-12-01/¨hd

User 3.1 O ***nej farvel***

transaction 3.1.1 --- Type:reject --- SubType:4 --- 2002-12-01/¨hd

transaction 3.1.2 --- Type:start --- SubType:5 --- 2002-12-01/¨hd

System 4.1 **- Farvel og tak for dit opkald.**

transaction 4.1.1 --- Type:success --- SubType:5 --- 2002-12-01/¨hd

Event 1 **nothing recorded**

Event 2 **disconnect**